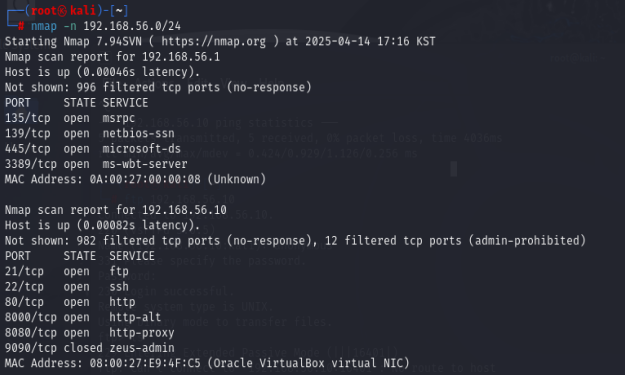
**1. 아이피 확인 및 포트 스캐닝**

nmap 192.168.56.0/24

대상자의 아이피가 192.168.56.10 인 것을 확인하였다.



nmap -A -p- -sS -sC -sV 192.168.56.10

21번, 22번, 80번, 7979번, 8000번, 8080번 포트 총 6개가 확인된다.

ftp는 anonymous가 허용된다고 한다.



**5. 8000번 포트**

**5-1. hanselandgretel**

이제 8000포트를 시도를 해보도록 한다.

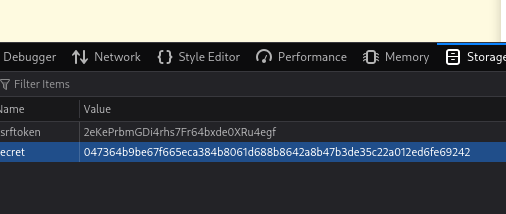
drib탐색 및 웹페이지 수동 탐색을 하면 아래 화면들을 만날 수 있다.



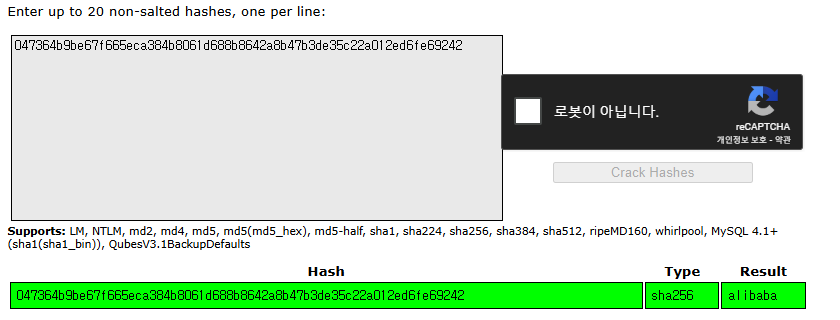


1. 첫번째 방법: 바로 확인하기

F12를 눌러 쿠키를 확인하면 secret에 암호화된 것 같은 단어들이 있다.



디코딩하면 alibaba가 나온다.



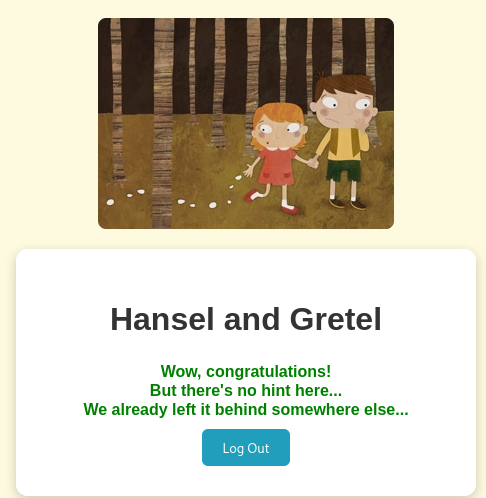
1. 두번째 방법: 로그인 후 힌트 보고 확인하기

hydra 192.168.56.10 http-form-post "/login/:username=^USER^&password=^PASS^:Login failed. Please try again." -l admin -P /usr/share/wordlists/rockyou.txt -s 8000

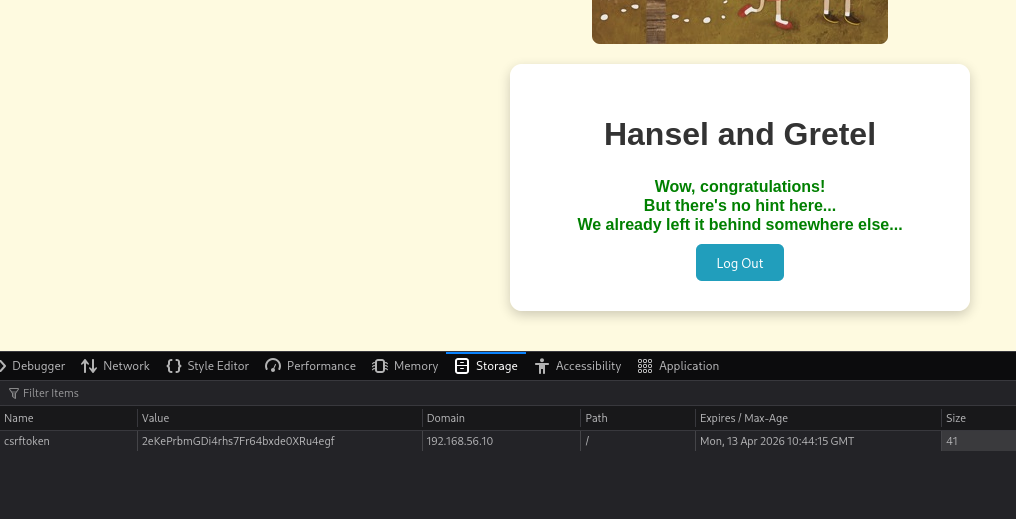
텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

Hydra로 id, password 획득 후 로그인하면 아래와 같은 화면이 나온다.



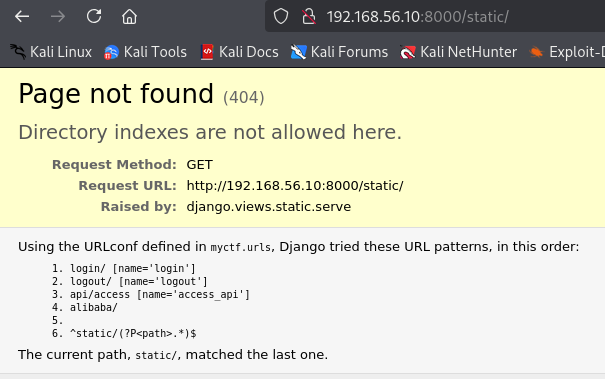
이미 힌트가 있었다고 한다.



위 이미지처럼 로그인한 후에는 쿠키가 보이지 않아 로그아웃하여 다시 돌아가서 쿠키를 확인 후 첫번째 방법과 동일한 절차 수행 후 alibaba라는 힌트를 얻는다.

1. 세번째 방법: dirb한 결과에서 확인하기

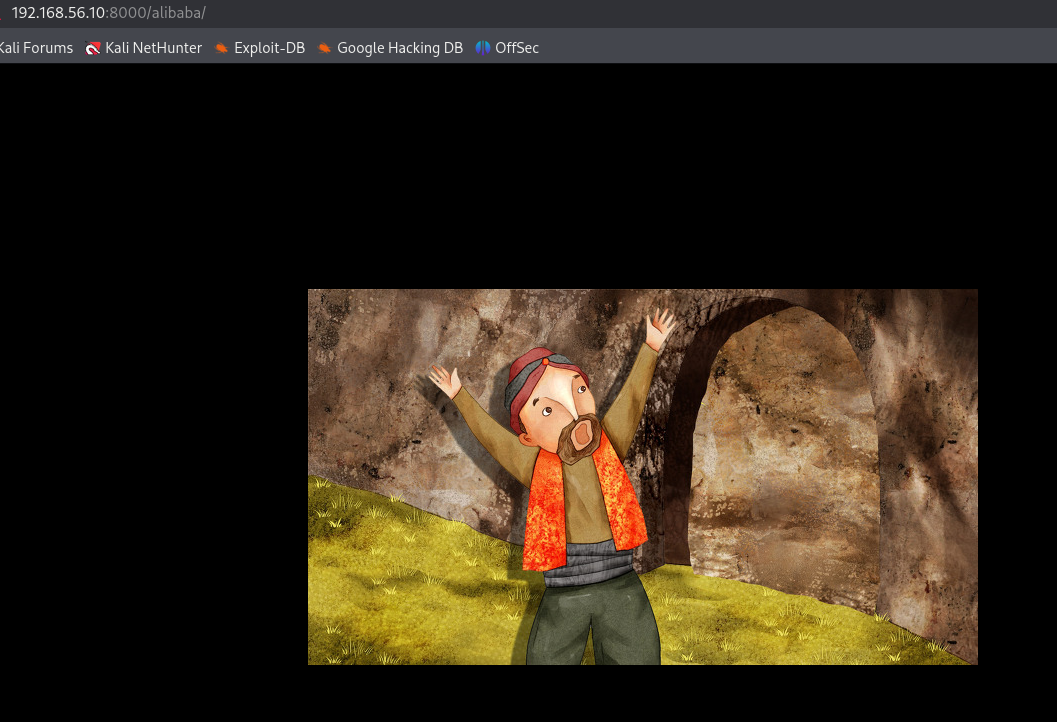
정보수집과정에서 dirb를 하면 static디렉토리를 발견할 수 있는데



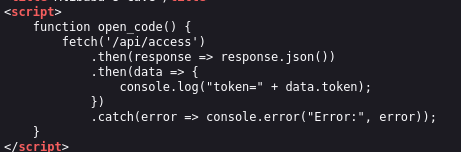
여기서 alibaba라는 디렉터리를 발견할 수 있다.

**5-2. alibaba**

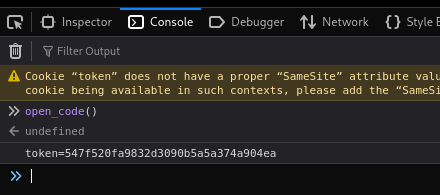
<http://192.168.56.10:8000/alibaba> 접속



1. 첫번째 방법: 페이지 소스를 확인한다.



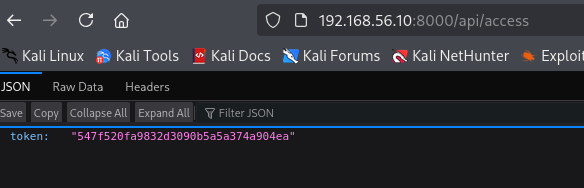
open\_code( )라는 함수가 있는 것을 확인할 수 있다.



위 이미지처럼 console에 입력하면 암호가 나온다.

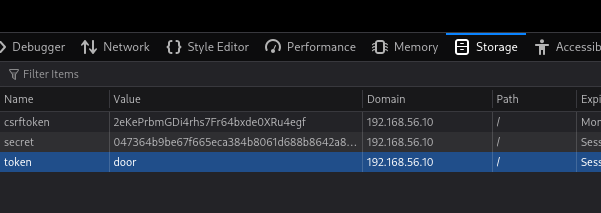
1. 두번째 방법: dirb에서 찾은 디렉터리 사용

<http://192.168.56.10:8000/static> 에 들어갔을 때 오류창에서 /api/access라는 디렉터리를 발견할 수 있다.



웹으로 접속하면 console에 입력한 것과 동일한 암호가 나온다.

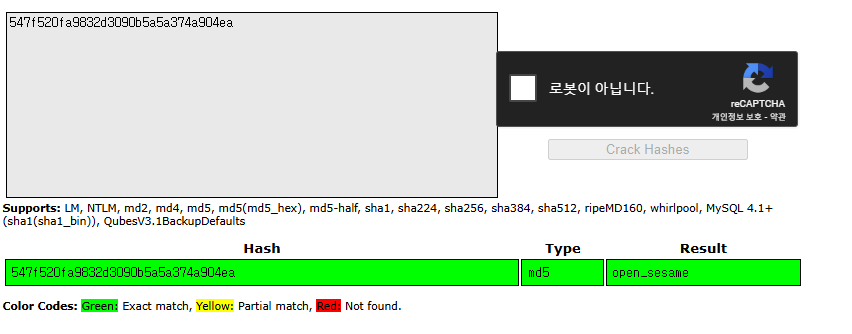
<http://192.168.56.10:8000/alibaba> 에서 f12를 누르면 아래 이미지처럼 나온다.



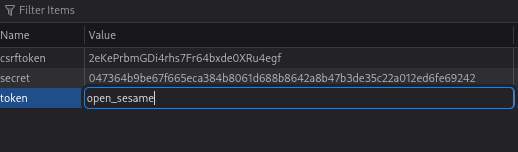
door 부분에 방금 얻은 암호를 넣는다.



새로고침하면 위 이미지처럼 문장이 나온다.



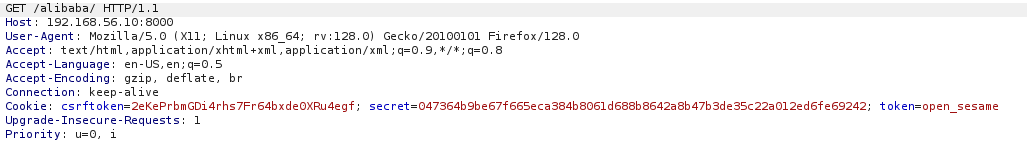
암호를 크랙하면 open\_sesame이라는 단어를 얻을 수 있다.



쿠키에 다시 open\_sesame을 넣고 새로고침하면



새로운 문장이 나왔다. 새로고침하여 Burp suite로 해당 페이지를 intercept한다.



GET을 POST로 바꾸고 forward 한다.





화면 밑에서 힌트 획득.

**10. gilhyeong 계정**

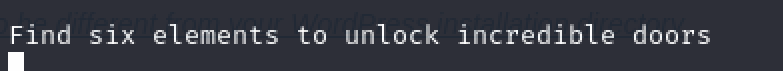
gilhyeong 접속 후

/home/gilhyeong에서 ls -al를 하면 .hint.txt를 찾을 수 있다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

.hint.txt의 내용



Find 이후에 단어들의 첫글자만 따면 find setuid로 setuid를 이용해 권한 상승을 하면 된다.

텍스트, 스크린샷, 폰트, 메뉴이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

저 중에서 기존에 존재하지 않는 파일은 netup, Sysmon, rootmon, fsrepair, netservicem, access-grant이다.

파일의 내용을 확인하기 위해 cat명령어를 사용하면

텍스트, 스크린샷, 폰트, 흑백이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

이런 식으로 읽기 힘든 내용이 나온다. 그래서 strings를 사용하여 내용을 확인한다. 전체 내용을 보기 위해 | more을 이용한다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

Strings를 사용하면 위와 같이 이미지가 나온다. 6개의 파일을 모두 strings로 확인해서 권한상승을 위해 setuid를 찾아야 한다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 폰트, 메뉴이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 메뉴, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 폰트, 메뉴이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 메뉴, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

각 파일을 실행해보자.

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 메뉴, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

명령어를 실행했을 때 출력된 내용과 strings 내용을 비교해서 확인해보면

/usr/bin/sysmon은 다른 명령어 없이 문장만 출력했다는 것을 알 수 있다.

/opt/bin/access-grant는 cat /root/access.txt를 보아 /root/access.txt의 파일 내용을 읽었을 가능성이 있다.

/usr/sbin/netservice는 ss -tulnp 명령어를 실행했을 가능성이 있다.

/usr/sbin/fsrepair는 echo 명령어를 통해 문장을 출력했음을 추측할 수 있다.

/usr/sbin/rootmon는 cat /root/system.info로 보아 /root/system.info의 파일 내용을 읽었을 가능성이 있다.

우리가 확인해봐야할 것은 /opt/bin/access-grant, /usr/sbin/netservice, /usr/sbin/fsrepair, /usr/sbin/rootmon이다.

우선 cat 명령어가 있는 파일부터 확인을 해보자.

텍스트, 스크린샷, 메뉴이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

권한 상승이 되지 않았다.

그럼 echo를 확인해보자.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

권한 상승이 되지 않았다.

마지막으로 ss를 확인해보자.

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

권한 상승 성공했다.

**11. root계정**

텍스트, 스크린샷이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

/ 경로에 …이라는 파일이 있다.

만약 cat 명령어가 실행되지 않는다면 chmod -x /tmp/cat을 한다. 이전 과정에서 PATH에 /tmp 경로를 추가하였기 때문에 /tmp/cat 명령어가 실행되어 본래의 cat명령어가 실행되지 않는 것이다.

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.